Structure Learning of Bayesian Network from the Data

Naveen Kumar Bhimagavni and T. Adilakshmi

Abstract Learning the structure of Bayesian network is a two-step process, one is parameter learning, and other is finding the best structure among search space using uncertain and incomplete data. Structure learning is the most important and complex task (NP hard problem) in estimation theory. However, existing techniques such as K3 algorithm require topological order of the nodes or a constraint on the maximum number of parents for a node, and also, most of them are generating all possible graphs even for small number of random variables and consume large amount of space and time complexity (Buntine in IEEE Trans Knowl Data Eng 8:195–210 [1]; Chickering in Learning from data. Springer, pp 121–130 [2]) to verify each of the structure. In this work, we propose an algorithm that generates comparatively small number of graphs as a heuristic search technique, and Bayesian score is calculated for each candidate structure to find the best network structure. The proposed algorithm consumes comparatively less time complexity to discover network structure using the data.

Keywords Bayesian network · Covid 19 dataset · Bayesian score · Subset graph algorithm

1 Introduction

Bayesian network is a compact representation of joined probability distribution and referred as directed acyclic graph (DAG) that consists of vertices and edges. Each vertex denotes a random variable, and edge from (A → B) represents the conditional dependency between random variables A and B. BN is a compact representation of joint probability distribution of all random variables [3].

N. K. Bhimagavni
University College of Engineering, Osmania University, Hyderabad, India

T. Adilakshmi (✉)
Vasavi College of Engineering, Osmania University, Hyderabad, India
e-mail: t_adilakshmi@staff.vce.ac.in

A. Kumar et al. (eds.), *Proceedings of the 2nd International Conference on Cognitive and Intelligent Computing*, Cognitive Science and Technology,
https://doi.org/10.1007/978-981-99-2742-5_35

1.1 Structure Learning

Bayesian network can be constructed manually by domain experts using prior structure knowledge especially in medical domain. The structure can then be refined manually or by automatically using techniques such as refinement algorithm and ExpertBayes [4]. However, manual construction may not be feasible for all domains. Another approach is to use available information. Structure of BN can also be estimated directly from the data. There are two approaches for the structure learning: score-based approach and constraint-based approach [5].

Constraint-Based Approach

The constraint-based case employs the independence test [6] to identify a set of edge constraints [7] for the graph and then finds the best DAG that satisfies the constraints [8].

Score-Based Approach

The score-based approach first defines a criterion to evaluate how well the Bayesian network fits the data and then searches over the space of DAGs [9] for a structure with maximal score [10].

Most of the existing techniques [1] follow the below steps to find the best structure. First, generate all possible graphs (which consumes large time complexity [2]) and then apply any scoring function for each candidate structure; the structure with the highest Bayesian score is considered as the best Bayesian network structure. However, the scoring-based approach generates exponent number of candidate graphs.

2 Proposed Algorithm

In this work, a new heuristic search technique is proposed that generates optimal number of graphs, and then, scoring function is applied on each of the graph to find the best structure.

Step 1: Generate optimal number of candidate structure.
Step 2: Calculate Bayesian score for each candidate structure using sufficient statistics of the input data.
Step 3: The structure with the highest Bayesian score is the best Bayesian network structure.

Algorithm 1: Proposed algorithm

```
Data:
   Input data for covid 19
Result:
   Bayesian Network structure
      1. begin
```

```
    2. Read Input data for covid 19;
    3. Best_Score = 0;
    4. Best_structure = 0;
    5. Calculate Sufficient statistics (S) for the input data;
  6. Call G = GenerateOptimalGraphs(n); // n is number of random
variables in the Input data
    7. for each graph g_i in G
begin
     8. Calculate Bayesian Score(BS(g_i))for each graph g_i using
Sufficient statistics (S);
    9. if BS(g_i) > Best_Score then
  Best_Score = BS(g_i);
  Best_structure = g_i
end
    10. end
```

3 Calculate Bayesian Score for Each Structure

Bayesian score [3] can be used as scoring function for all possible graphs [11, 12]. It is the function which takes one graph as an input and computes the Bayesian score for that structure [13, 14].

$$P\left(\frac{G}{D}\right) = \frac{P\left(\frac{D}{G}\right)P(G)}{P(D)} \tag{1}$$

$P(D \mid G)$ is the marginal likelihood of the graph structure G. $P(G)$ is the prior probability of the graph structure and can be considered as uniform distribution G in the candidate structure space. $P(D)$ is the probability of the data which acts as a normalizing constants the score function can be written as below; it primarily depends on the marginal likelihood $P(D \mid G)$

$$\text{ScoreB}(G : D) = \log P\left(\frac{D}{G}\right) \tag{2}$$

3.1 Data Preprocessing

The data was collected from source kaggle [15] for the pandemic disease Covid-19 that consists of 35 million records; the dataset contains 27 features with missing values. These values are handled using Python package. Test data contains missing

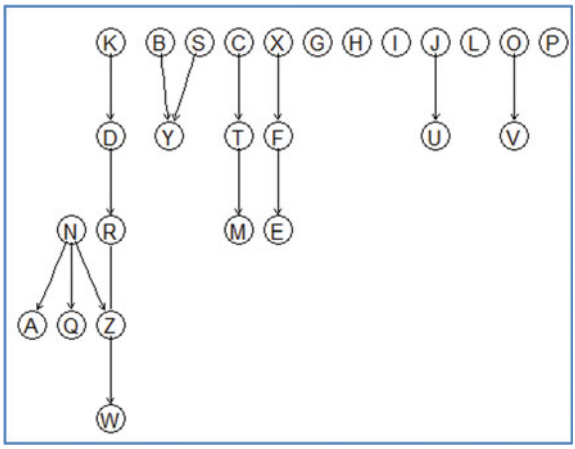

Fig. 1 Best Bayesian network structure for covid_19 dataset created by proposed algorithm

values for mixed data types (numeric or strings), and SimpleImputer from the Python package sklearn.impute can be used.

Ex: imp_mean = SimpleImputer (missing_values=np.nan, strategy="most_frequent")

When strategy is set to "most_frequent", it replaces missing values using the most frequent value along each column. It can be used with strings or numeric data (mixed data types) (Fig. 1).

4 Goodness Fit of the Model

Proposed heuristic algorithm can be compared with the existing structure learning algorithms such as hill-climbing learnt by bnlearn package. Bayesian network models can be compared with the following metrics.

1. score comparison
2. hold-out cross-validation
3. precision.

1. Score Comparison

The Bayesian scoring function is used to compute the score of the Bayesian network with 27 nodes learnt using bnlearn package (hill-climbing approach)

$$\text{score(bnlearn\_BN, 26 nodes)} = 3240.513$$

Score of the Bayesian network with 20 nodes learnt using heuristic algorithm

$$\text{score(heuristic\_BN, 26 nodes)} = 3576.921$$

2. Hold-Out Cross-Validation

Cross-validation is a standard method used to estimate a model's goodness of fit. In k-fold cross-validation, data is randomly divided into k subsets. For each subset k, a model is evaluated on k having been trained on $(k - 1)$ subsets (Fig. 2; Tables 1 and 2).

3. Precision Analysis

Precision: It computes the proportion of positive identifications was actually correct out of total predicted positives. 3×3 confusion matrix created for the node sore throat which has three levels, namely low, moderate and high, diagonal elements represent the true positives along that column, and the remaining values in the same column represent false positives. Precision values have been computed with various

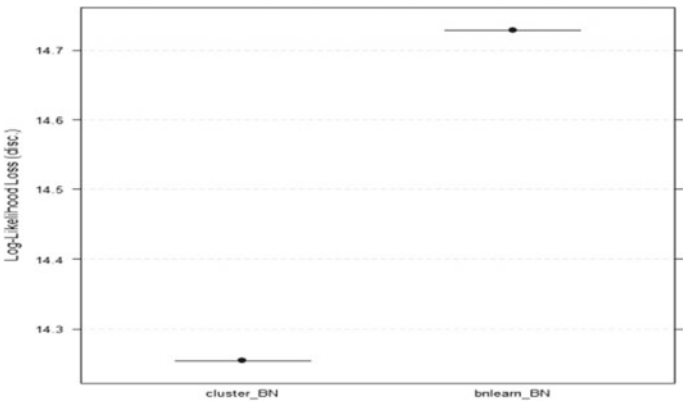

Fig. 2 Expected loss comparison graph for bnlearn_BN and heuristic_BN

Table 1 Expected loss for the Bayesian network created by bnlearn package

Number of folds in cross-validation (k)	$K = 10$
Number of observations that are to be sampled for the test subsample	$m = 50{,}000$
Loss function	Log-likelihood loss
Expected loss	14.80372

Table 2 Expected loss for the Bayesian network created by heuristic algorithm

Number of folds in cross-validation (k)	$K = 10$
Number of observations that are to be sampled for the test subsample	$m = 20{,}000$
Loss function	Log-likelihood loss
Expected loss	14.65423

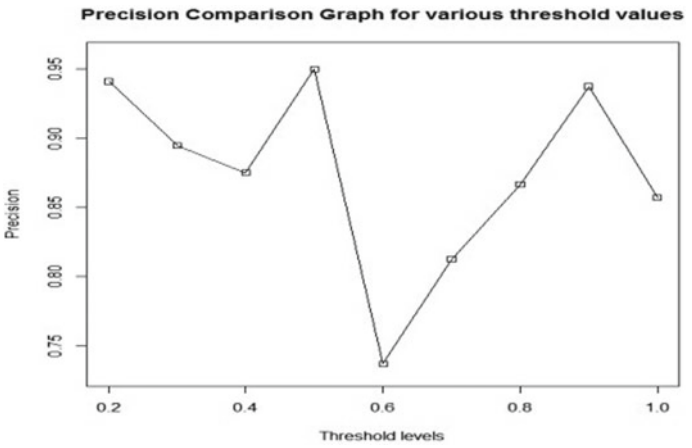

Fig. 3 Precision values computed for different threshold values (0.2–1.0)

Table 3 3 × 3 confusion matrix created for the node sore throat

Observed values	Predicted values		
	Low	Moderate	High
Low	0	1	1
Moderate	0	8	0
High	0	0	4

threshold values ranging from 0.2 to 1.0 for the below confusion matrices from precision comparison graph shown in Fig. 3 (Table 3).

5 Conclusion

The proposed algorithm generates less number of candidate structures, thereby consumes comparatively less time complexity and finds the best structure in optimal time. Bayesian score is used as a scoring function to compute the score of the each candidate structure. The goodness fit of the best Bayesian network structure is measured in terms of Bayesian score, expected loss and precision values. It is observed that score of the best Bayesian network structure has improved by 10%. At the same time, expected loss is reduced by 1.0098 percentages.

However, present work can be extended for large number of nodes in the graph, and Bayesian score can be calculated for Bayesian networks having both discrete and continuous random variables.

References

1. Buntine W (1996) A guide to the literature on learning probabilistic networks from data. IEEE Trans Knowl Data Eng 8(2):195–210
2. Chickering DM (1996) Learning Bayesian networks is NP-complete. In: Learning from data. Springer, pp 121–130
3. Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press
4. Beretta S, Castelli M, Gonçalves I, Henriques R, Ramazzotti D. Learning the structure of Bayesian networks: a quantitative assessment of the effect of different algorithmic schemes
5. Bernstein D, Saeed B, Squires C, Uhler C (2020, June) Ordering-based causal structure learning in the presence of latent variables. In: International conference on artificial intelligence and statistics, pp 4098–4108. PMLR
6. Chen Y, Tian J (2014, June) Finding the k-best equivalence classes of Bayesian network structures for model averaging. Proc AAAI Conf Artif Intell 28(1)
7. Chen EYJ, Shen Y, Choi A, Darwiche A (2016) Learning Bayesian networks with ancestral constraints. Adv Neural Inf Process Syst 29:2325–2333
8. Cheng J, Greiner R, Kelly J, Bell D, Liu W (2002) Learning Bayesian networks from data: an information-theory based approach. Artif Intell 137(1–2):43–90
9. Chickering D (2002) Learning equivalence classes of Bayesian-network structures. J Mach Learn Res 2:445–498
10. Chickering DM, Meek C (2002, August) Finding optimal Bayesian networks. In: Proceedings of the eighteenth conference on uncertainty in artificial intelligence, pp 94–102
11. Chobtham K, Constantinou AC (2020) Bayesian network structure learning with causal effects in the presence of latent variables. In: Proceedings of the 10th international conference on probabilistic graphical models, in proceedings of machine learning research, vol 138, pp 101–112
12. Claassen T, Mooij JM, Heskes T (2013, August) Learning sparse causal models is not NP-hard. In: Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence, pp 172–181
13. Colombo D, Maathuis MH (2014) Order-independent constraint-based causal structure learning. J Mach Learn Res 15(1):3741–3782
14. Colombo D, Maathuis MH, Kalisch M, Richardson TS (2012) Learning high-dimensional directed acyclic graphs with latent and selection variables. Ann Stat:294–321
15. Kaggle data source for Covid19. https://www.kaggle.com/iamhungundji/covid19-symptoms-checker